

Список цитируемых источников

1. Тенденции развития техники [Электронный ресурс]. — Режим доступа: <https://www.km.ru/referats/94C44F0432A04C2CA8B85-FD154C81866>. — Дата доступа: 15.04.2023.
2. Тенденции развития информационных технологий [Электронный ресурс]. — Режим доступа: <https://www.vsavm.by/knigi/kniga3/1770.html>. — Дата доступа: 15.04.2023.
3. Основные тенденции в области науки и техники [Электронный ресурс]. — Режим доступа: <http://eurasian-defence.ru/?q=node/37689>. — Дата доступа: 15.04.2023.

УДК 004.43

А. В. Шах

*Учреждение образования «Барановичский государственный университет», Барановичи,
Республика Беларусь*

ЯЗЫКИ ПРОГРАММИРОВАНИЯ ПРОМЫШЛЕННЫХ РОБОТОВ

Введение. Разработка и внедрение промышленных роботов имеют большую актуальность в современном мире, так как они представляют собой эффективный способ автоматизации производства, что позволяет повысить производительность, улучшить качество продукции и снизить затраты на трудоемкую работу.

Промышленные роботы также способны выполнять тяжелые и опасные задачи, такие как сварка или работа с химическими веществами, что повышает безопасность рабочей среды для человека.

В связи с быстро меняющимся рынком и повышением требований к производительности и качеству продукции, внедрение робототехники позволяет компаниям сохранять конкурентоспособность и эффективность. Кроме того, промышленные роботы могут быть использованы в различных отраслях, таких как автомобильная, электронная, медицинская и другие, что позволяет сократить время производства и улучшить качество продукции в различных отраслях.

Разработка и реализация робота — это два разных процесса. Для того, чтобы робот мог выполнять задачи, которые очень похожи на то, что делает человек, необходимы специальные языки программирования для роботов. Они позволяют роботам иметь «интеллект» и возможность выполнения важных задач.

На рынке существует огромное количество языков программирования, но в робототехнике наиболее распространены C++ и Python, которые пользуются популярностью благодаря своей функциональности и простоте использования. Однако, для создания более адаптированных роботизированных операционных систем, используемых ведущими производителями промышленных роботов, могут использоваться более старые и устаревшие языки, такие как Pascal, которые могут служить основой для проприетарных языков.

Основная часть. Проприетарный язык программирования — это язык, который разработан и поддерживается определенной компанией, и его использование и распространение могут быть ограничены правами интеллектуальной собственности. Он может использоваться для разработки программного обеспечения, связанного с конкретными продуктами этой компании, такими как программное обеспечение для промышленных роботов, автомобилей или других устройств.

Языки программирования, такие как C, C++ и Python, широко используются в индустрии робототехники для создания роботов, обладающих различными функциями, от двигательных функций до искусственного интеллекта, необходимого для роботов. Эти языки программирования имеют обширную сеть пользователей, которые могут помочь инженерам в их работе.

С увеличением рынка роботов для промышленности, растет и спрос на квалифицированных разработчиков и инженеров, владеющих языками C, C++ и Python.

Хотя инженеры, занимающиеся различными типами роботов, могут использовать комбинацию популярных языков программирования для создания полностью функциональных роботов, ведущие компании по производству промышленных роботов, такие как ABB, Comau и Fanuc, предпочитают использовать собственные проприетарные языки программирования, которые их инженеры изучают на работе. Многие из этих проприетарных языков программирования основаны на Паскале, что делает его популярным языком для будущих инженеров.

Независимо от выбранного языка программирования для работы с роботами, полученные знания будут полезны в других робототехнических проектах. В дополнение к C, C++ и Python, существуют и другие языки программирования, которые полезны для работы с роботами и которые каждый начинающий программист и разработчик должен изучить.

Оба языка программирования, C и C++, часто считают золотым стандартом в мире программирования. Оба являются зрелыми языками общего назначения и обеспечивают производительность в реальном времени, то есть запрограммированные действия выполняются точно в указанное время. В робототехнике это особенно важно, поскольку роботы должны выполнить задачи, когда этого требуют люди [1].

Хотя С и С++ обычно требуют написания большего количества кода и не так удобны для пользователя, как другие языки программирования, например Python, оба языка программирования имеют явное преимущество в том, что многие аппаратные библиотеки используют С или С++ в своих собственных разработках. Это повышает совместимость этих языков с критическим оборудованием, что важно в робототехнике. Кроме того, С++ является более усовершенствованной версией С, которая поддерживает объекты и классы.

С# — это язык программирования, разработанный компанией Microsoft и используемый в программе Microsoft Visual C# Express для разработки программного обеспечения в среде .NET. Он часто используется для создания сетевых и коммуникационных приложений, таких как порты и сокет, что делает его идеальным выбором для интеграции нейронных сетей в роботизированные приложения.

Языки описания оборудования (HDL) являются важным инструментом для программирования роботов, а Verilog и язык описания оборудования VHSIC (или VHDL) упрощают этот процесс. Однако, для работы с языками описания оборудования необходим хороший опыт работы с электроникой.

Java — это еще один популярный язык программирования, который используется для разработки программного обеспечения промышленных роботов. Это интерпретируемый язык программирования, который не встроен в машинный код и может быть выполнен на разных компьютерах благодаря виртуальной машине Java. Он широко используется в области искусственного интеллекта и позволяет создавать нейронные сети. Java даже использовался для программирования суперкомпьютера IBM Watson, который помогает предприятиям интегрировать искусственный интеллект в свою деятельность [2].

Lisp — один из старейших языков программирования, который широко используется в робототехнике. Lisp легок в использовании и в настоящее время воскрешается для использования в создании искусственного интеллекта и роботизированных приложений. Большая часть роботизированных операционных систем, которые используются для разработки роботизированных приложений, написана на Lisp. Lisp имеет множество преимуществ, которые делают его легким в использовании, включая структуры данных в виде деревьев или нелинейных структур, автоматизированное управление памятью и подсветку синтаксиса [3].

MATLAB — отличный инструмент для разработчиков, которые хотят анализировать данные, запускать симуляции и создавать системы управления роботами с помощью специально разработанных интерфейсов. С помощью Robotics Toolbox for MATLAB, который включает в себя функции для кинематики, динамики и генерации траекторий, разработчики могут создавать системы для исследования и моделирования роботов-манипуляторов [4].

Pascal является источником для многих проприетарных языков программирования, но очень мало роботов программируется на этом языке. Несмотря на это, многие производители промышленных роботов, такие как ABB и Kuka, используют Pascal в качестве основы для своих собственных языков программирования. Pascal считается отличным способом для новых разработчиков понять основные функции и технические детали проприетарных языков программирования и получить возможность изучить передовой опыт программирования роботов.

Python — один из самых простых языков программирования, который также является одним из самых популярных, наряду с С и С++. Так как Python является интерпретируемым языком, он обладает обширным набором библиотек, что упрощает реализацию основных функций. Кроме того, Python требует меньше строк кода, что делает его быстрее и проще в использовании и изучении по сравнению с С, С++ и Java. Python сокращает время программирования, устраняя необходимость определять и приводить типы переменных, а также позволяет кодировать движения в одном скрипте. Благодаря широкому распространению и популярности, Python также имеет большое сообщество разработчиков, которое может стать отличным ресурсом для новичков в процессе обучения.

Заключение. При выборе языка программирования для промышленных роботов необходимо учитывать несколько факторов:

- нужно рассмотреть тип и модель робота, который будет программироваться. Некоторые производители роботов могут предпочитать использовать определенные языки программирования или даже предоставлять свои собственные языки.
- следует учитывать опыт и уровень знаний разработчика. Некоторые языки программирования могут быть проще в использовании и быстрее в изучении, особенно для новичков.
- стоит рассмотреть функциональные требования для конкретного проекта, такие как типы датчиков и механизмы управления движениями. Некоторые языки программирования могут лучше подходить для определенных типов задач.
- необходимо учитывать доступность и поддержку языка программирования. Некоторые языки могут иметь большое сообщество разработчиков и библиотек, что облегчает разработку и обеспечивает большую гибкость.

В целом, выбор языка программирования для промышленных роботов должен быть основан на конкретных требованиях проекта, уровне опыта разработчика и доступности языка.

Список цитируемых источников

1. Справочник по языку С++ [Электронный ресурс]. — Режим доступа: <https://learn.microsoft.com/ru-ru/cpp/cpp/cpp-language-reference?view=msvc-170>. — Дата доступа: 02.05.2023.

2. Java [Электронный ресурс]. — Режим доступа: <https://www.java.com/ru/>. — Дата доступа: 02.05.2023.
3. Common Lisp [Электронный ресурс]. — Режим доступа: <https://lisp-lang.org/>. — Дата доступа: 02.05.2023.
4. MATLAB and Simulink [Электронный ресурс]. — Режим доступа: <https://www.mathworks.com/>. — Дата доступа: 02.05.2023.

УДК 004.054

Н. А. Шустол¹, Е. Г. Шапович¹, И. М. Михеев²

¹ Учреждение образования «Барановичский государственный университет», Барановичи, Республика Беларусь

² Государственное бюджетное профессиональное образовательное учреждение «Северный техникум транспорта и технологий», Архангельск, Российская Федерация

РАЗРАБОТКА ПРИЛОЖЕНИЯ “OSKAR VIRTUAL ASSISTANT”

Введение. Виртуальные помощники значительно облегчили жизнь современного пользователя. Благодаря таким ассистентам как Алиса, Сири, Маруся, Google Assistant, пользователь может одной лишь командой поставить будильник, написать заметку, выполнить поисковой запрос, узнать рецепт приготовления блюда, а если ассистент использует трансформеры (например, GPT), то и найти себе неплохого собеседника. Однако разработка такого ассистента задача не из простых. В целях изучения данного вопроса, было принято решение по разработке собственного виртуального ассистента “Oskar Virtual Assistant” [1].

Основная часть. Виртуальный помощник — это приложение, которое понимает голосовые команды и выполняет задачи за пользователя [2]. Виртуальные ассистенты обрели большую популярность. Разработка виртуального помощника выгодна и в будущем может окупиться, так как автоматизация и быстрое выполнение рутинных задач с помощью искусственного интеллекта и голосового ввода приводит к экономии времени и денег пользователей. Поэтому в данный момент мы ведём активную разработку собственного ассистента, который будет выпущен в Google Play Market под издательством CLG Publisher [3]. На момент апреля 2023 года в магазине имеется черновая версия, но она не включает в себя последние наработки. Выход полноценной версии для тестирования планируется в ближайшее время.

История виртуальных помощников берёт своё начало с наработок Алана Тьюринга, который был великим ученым и пионером в области информационных технологий. Его работа была революционной и оказала огромное влияние на современную науку и технологии. Одним из наиболее знаменитых достижений Тьюринга было создание машины Тьюринга — универсальной машины, которая могла выполнять любые вычисления, основанные на математических алгоритмах. Это был первый шаг к созданию искусственного интеллекта. Алан Тьюринг вдохновил команду “Oskar Virtual Assistant” своими разработками и теперь мы стараемся сделать искусственный интеллект для нашего ассистента. Помощники с искусственным интеллектом призваны максимально эффективно помогать людям выполнять поставленные задачи.

“Oskar Virtual Assistant” — это виртуальный помощник, использующий искусственный интеллект, который может помочь во многих ситуациях. Он может отвечать на вопросы, помогать с поиском информации, предоставлять рекомендации и многое другое. Но самое главное, что делает Оскара уникальным, это то, что он может «понимать». Использование искусственного интеллекта позволяет Оскару учиться и развиваться. Да, многие приложения на смартфонах уже используют искусственный интеллект для решения задач и обработки данных. Например, голосовые помощники, переводчики, рекомендательные системы, фильтры спама и многие другие уже используют алгоритмы машинного обучения и нейронные сети для распознавания образов, классификации данных, прогнозирования результатов и улучшения пользовательского опыта. Развитие искусственного интеллекта и его интеграция в мобильные технологии будет продолжаться и в будущем. Мы создали алгоритм, который может понимать человека и обучаться. Этот алгоритм основан на использовании слов и их значение — смысл. Алгоритм анализирует запрос человека и начинает извлекать из запроса смысл, после чего и начинает писать ответ. У Оскара есть своя личная база данных — DB_Knowledge, которая предоставляет ему информацию для ответов на запросы пользователя.

Приложение разрабатывается в Android Studio на языке Java. На рисунке 1 можно увидеть внешний вид окна чата с ассистентом и пример того, как он отвечает на запросы.

Главная отличительная черта Оскара от других ассистентов — это персональность обучения. Очень хорошо, когда есть общая база данных между всеми пользователями и обучение нейросети проходит быстрее. Однако Оскар — это персональный друг, которого вы сможете обучить под себя. Ему легче будет понимать, что конкретно вы хотите от него, ему легче будет адаптироваться именно под вашу манеру построения запросов. По этой причине вся нейросеть находится в самом приложении, а база данных — на вашем устройстве.

Принцип работы алгоритмов Оскара:

1. Обработчик текста получает ввод пользователя и проводит первичную обработку текста.
2. Полученный текст передается в базу данных DB_Knowledge, где происходит поиск соответствующей информации.
3. Если запрос пользователя является математическим, активируется алгоритм «калькулятора» для выполнения расчетов.